Learning an Adaptive Forwarding Strategy for Mobile Wireless Networks: Resource Usage vs. Latency

Victoria Manfredi, Alicia Wolfe, Xiaolan Zhang, Bing Wang

Reinforcement Learning for Real Life Workshop @ NeurIPS 2022

Mobile wireless networks

Applications include

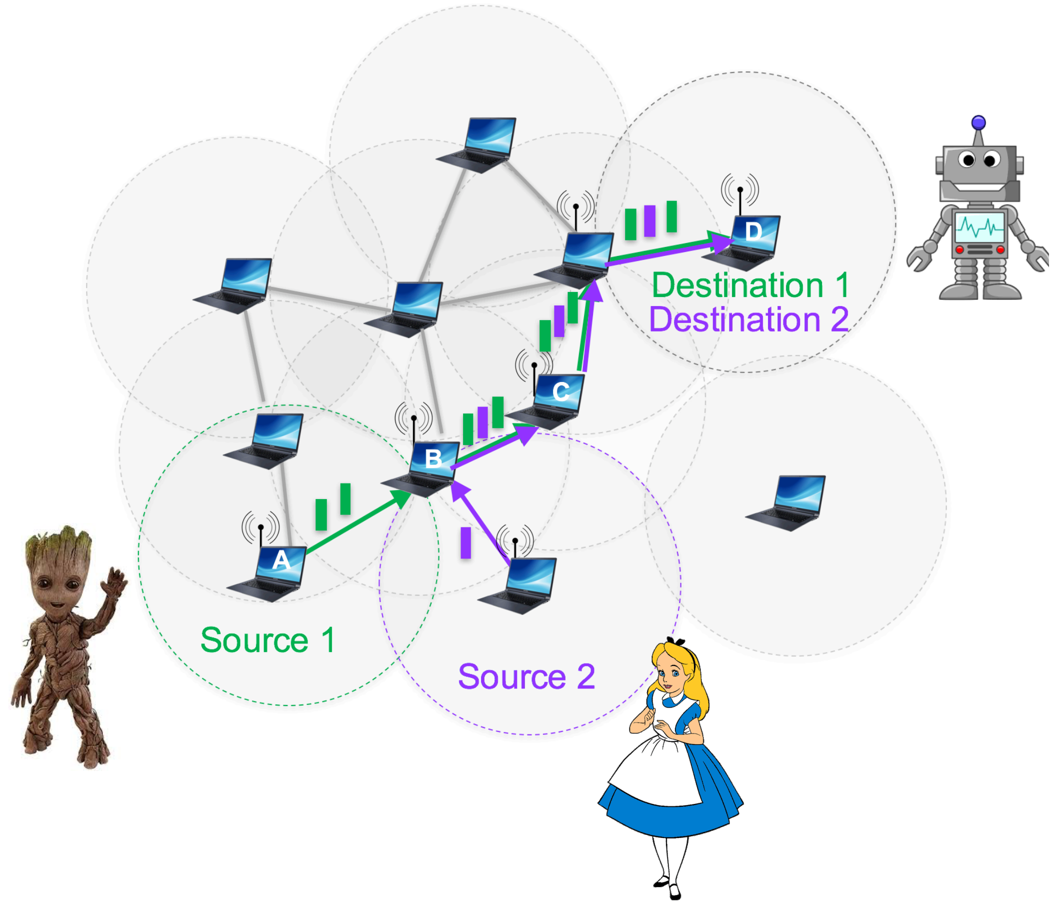
- vehicular safety, animal tracking, search-and-rescue, military …

Devices are moving

- lack of infrastructure simplifies deployment
- devices generate and forward traffic

Problem: forwarding traffic is hard

- **changing network conditions**: mobility, interference, traffic …
- **limited communication windows:** difficult to predict
- **decentralized architecture**: online training is difficult
- **competing network goals**: throughput, delay, resource usage



Mobile wireless networks

Applications include

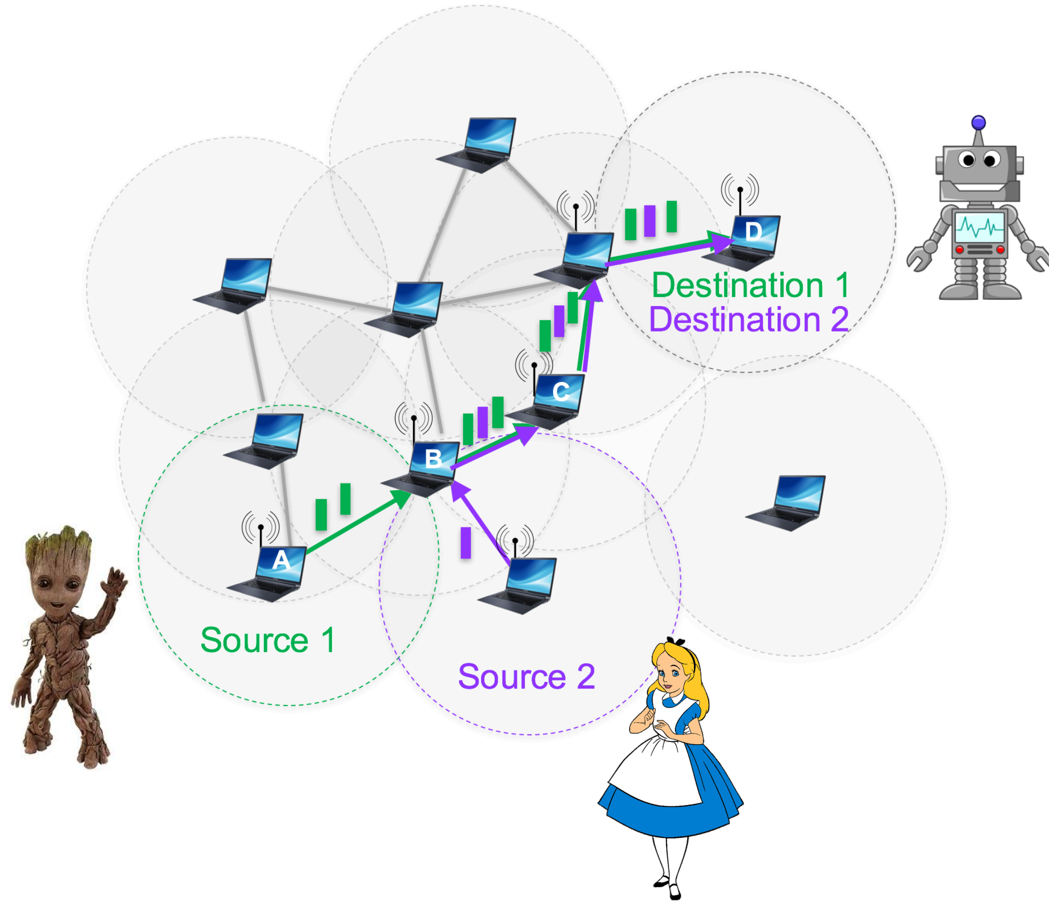
- vehicular safety, animal tracking, search-and-rescue, military …

Devices are moving

- lack of infrastructure simplifies deployment
- devices generate and forward traffic

Problem: forwarding traffic is hard

- **changing network conditions**: mobility, interference, traffic …
- **limited communication windows:** difficult to predict
- **decentralized architecture**: online training is difficult
- **competing network goals**: throughput, delay, resource usage



Solution: learn how to forward using deep reinforcement learning

Key ideas

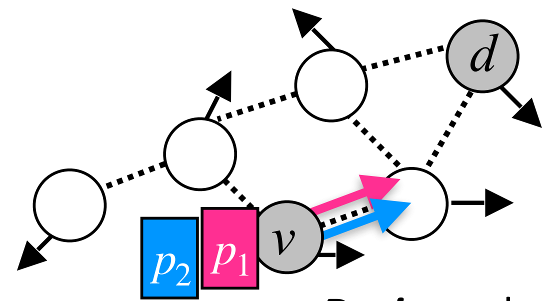
1. Packet-centric decisions

2. Independent decisions

3. Relational features

4. Weighted reward function

Problem: Normally a device chooses a packet's next hop … but a device's state doesn't track what happens to the packet

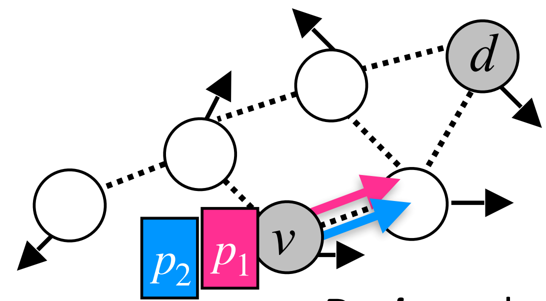


Device v chooses next hop for each outgoing packet

Key ideas

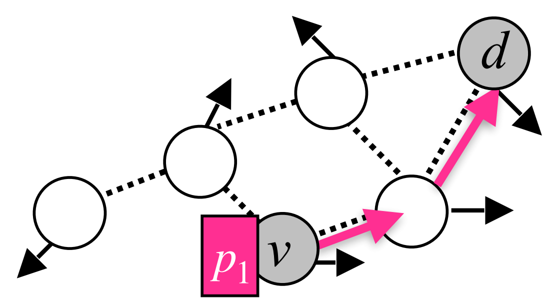
1. Packet-centric decisions
2. Independent decisions
3. Relational features
4. Weighted reward function

Problem: Normally a device chooses a packet's next hop … but a device's state doesn't track what happens to the packet



Device v chooses next hop for each outgoing packet

Solution: Use **packet agents** to simplify s, a, s', r experience sequence and define reward



Packet p_1 chooses its next hop at each device

Key ideas

Problem: Mobile network connectivity varies, but # of inputs to DNN are fixed

1. Packet-centric decisions
2. Independent decisions
3. Relational features
4. Weighted reward function

Key ideas

1. Packet-centric decisions

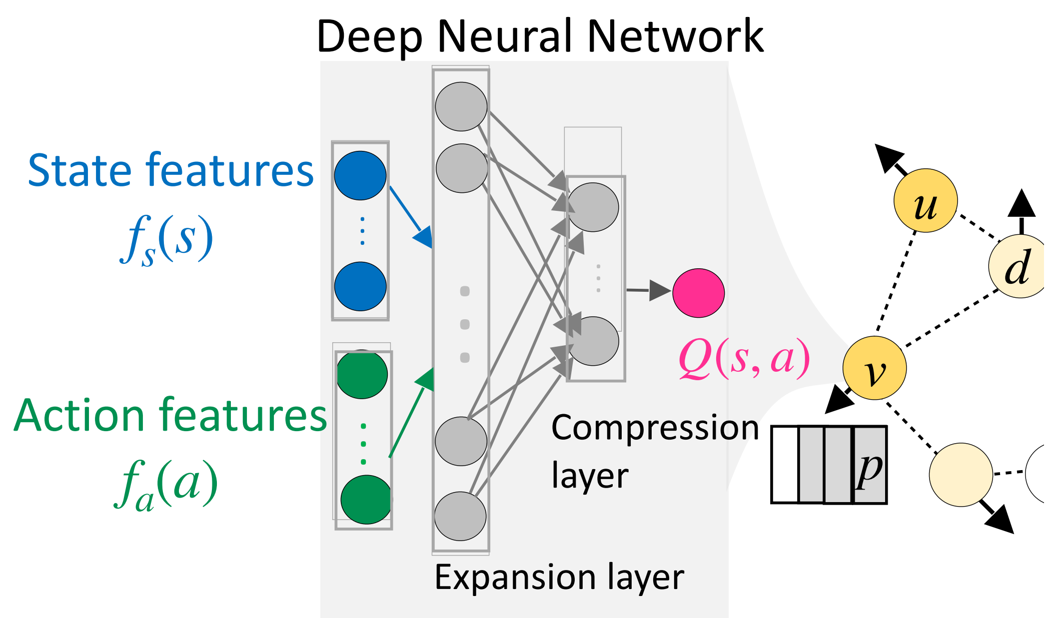
2. Independent decisions

3. Relational features

4. Weighted reward function

Problem: Mobile network connectivity varies, but # of inputs to DNN are fixed

Solution: Use DNN separately to predict Q-value for ***each action*** available in a state



How packet p at device v chooses next hop

- For each (state, action) pair, inputs features into DNN to get Q-value
- p chooses action that gives highest Q-value

Key ideas

1. Packet-centric decisions
2. Independent decisions
3. Relational features
4. Weighted reward function

Problem: How to define generalizable states and actions?

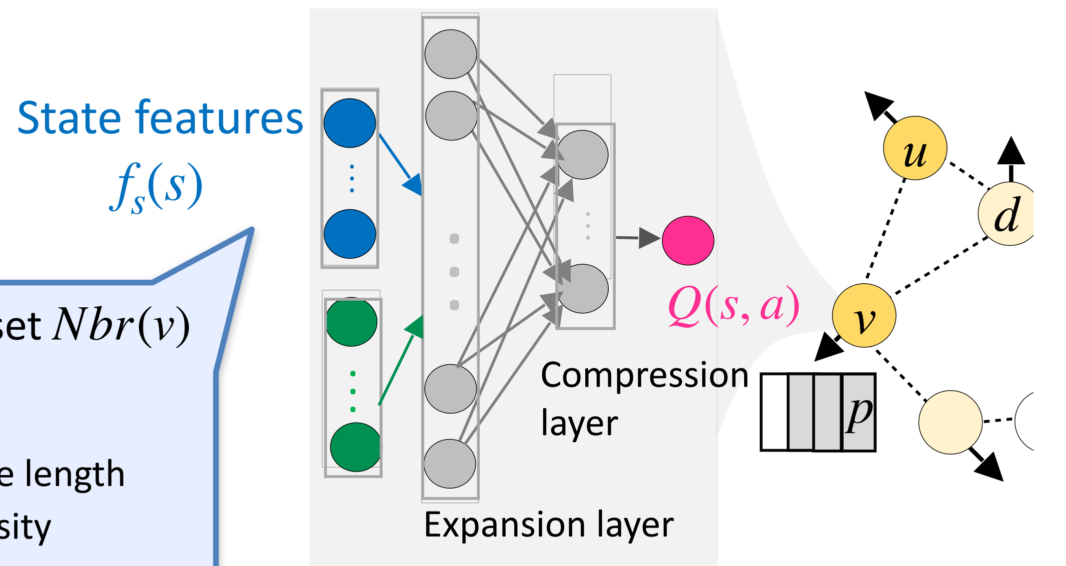
Key ideas

Problem: How to define generalizable states and actions?

1. Packet-centric decisions

Solution: Use **relational features** that model the relationship between devices instead of describing a specific device

2. Independent decisions



3. Relational features

For packet p at device v with 1-hop neighbor set $Nbr(v)$

Packet features $f_{packet}(p)$: p's TTL

Device features $f_{device}(v, d)$: v's queue length, queue length for only packets to p's dest., node degree, node density

Neighbor features, $f_{neighbor}(Nbr(v), p, t)$

- summarize varying # of neighbors
- min, mean, max of $f_{device}(Nbr(v), p, t)$

Path features $f_{path}(v, d)$: distance from v to d

Key ideas

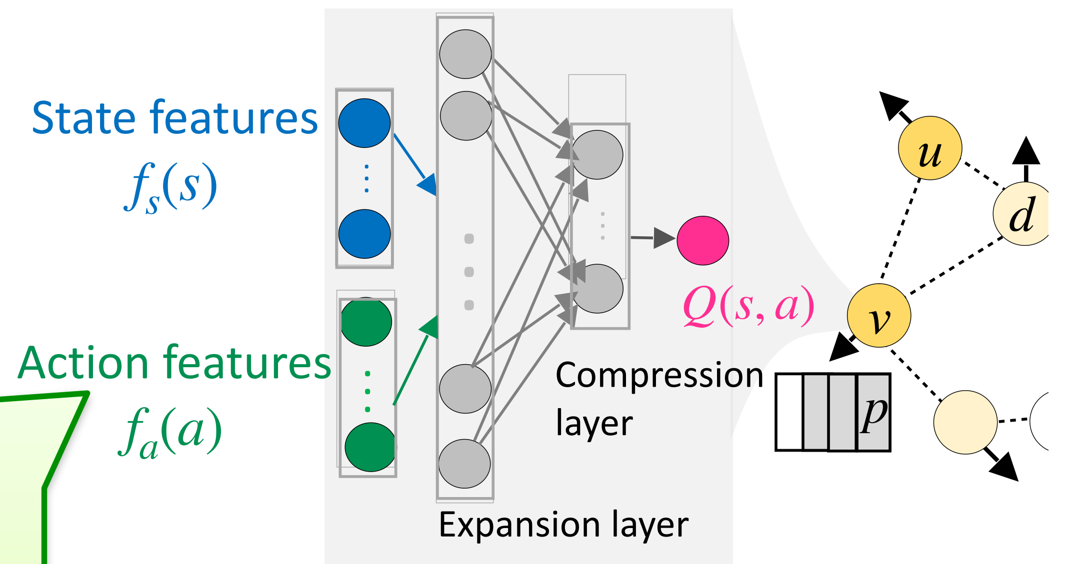
1. Packet-centric decisions

2. Independent decisions

3. Relational features

Problem: How to define generalizable states and actions?

Solution: Use **relational features** that model the relationship between devices instead of describing a specific device



Packet p separately considers each action u

- **device features** $f_{device}(u,d)$,
- **neighborhood features** $f_{nbrhood}(u,d)$,
- **device features** $f_{device}(u,d)$, and
- **context features** $f_{context}(p,u)$ which indicate whether p has recently visited u

Key ideas

1. Packet-centric decisions

2. Independent decisions

3. Relational features

4. Weighted reward function

Problem: How to trade-off competing network goals such as packet delay vs. resource usage?

Key ideas

1. Packet-centric decisions

2. Independent decisions

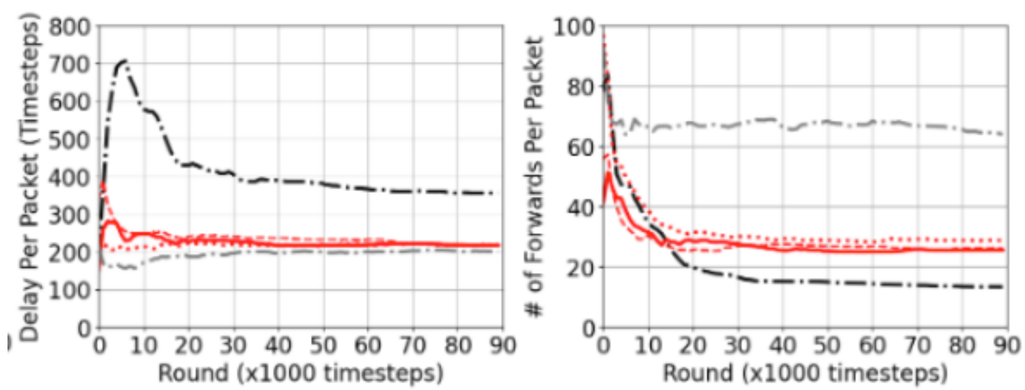
3. Relational features

4. Weighted reward function

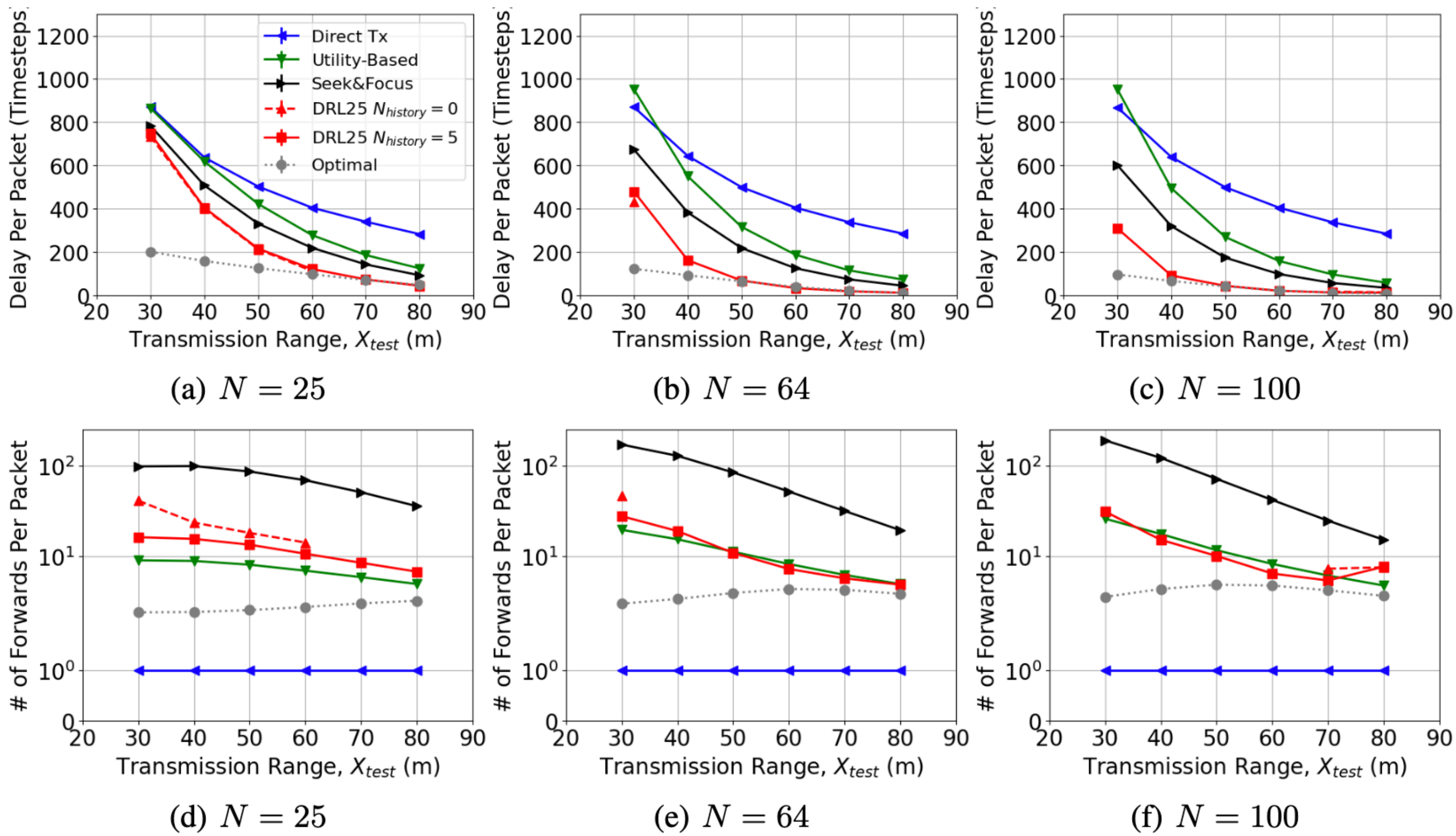
Problem: How to trade-off competing network goals such as packet delay vs. resource usage?

Solution: Use a weighted reward function

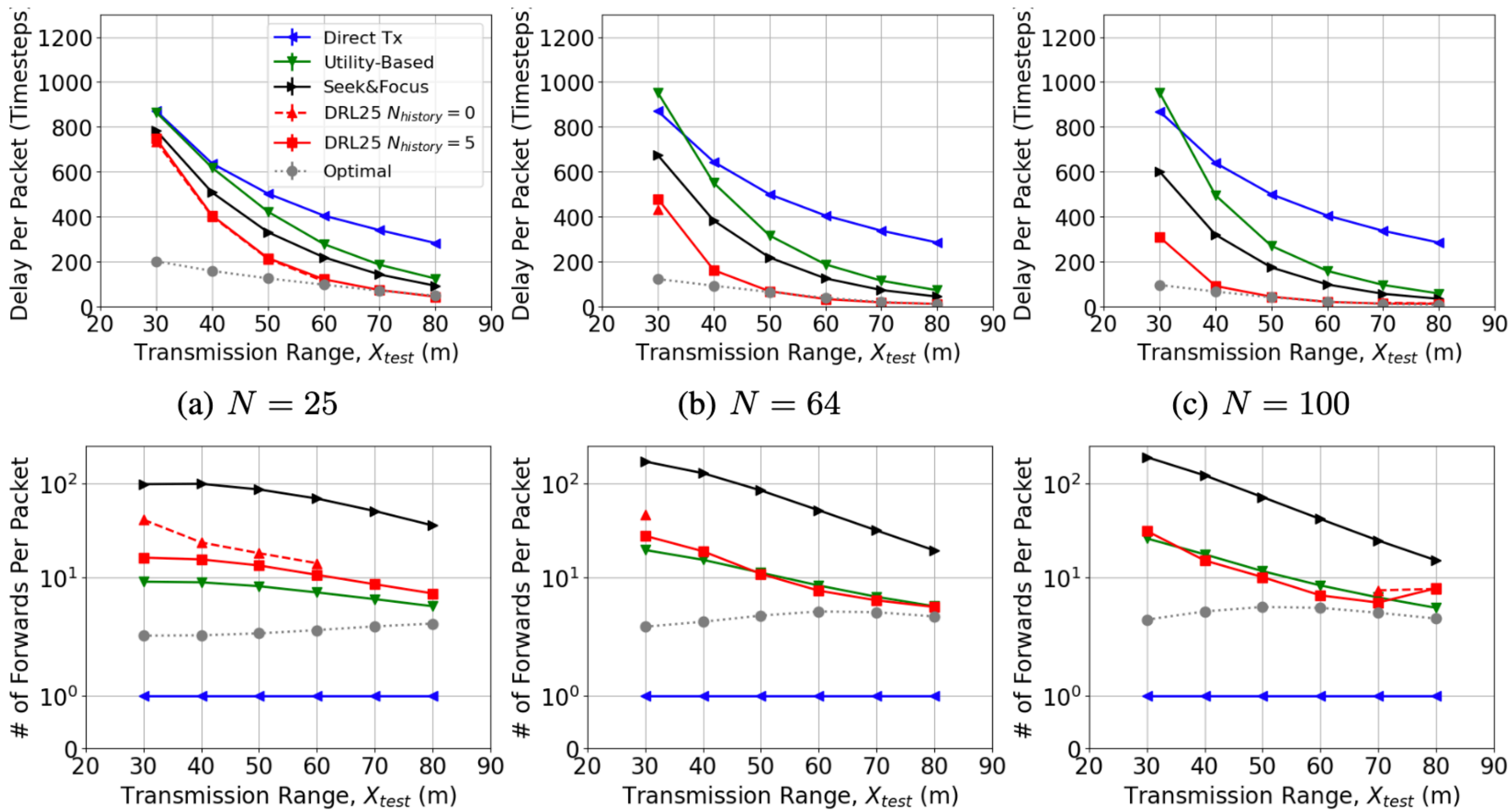
$$r_{transmit} = \alpha r_{stay},\ r_{delivery} = 0,\ r_{drop} = \frac{r_{transmit}}{1-\gamma}$$



Generalization is possible!



Generalization is possible!



(a) $N = 25$ (b) $N = 64$ (c) $N = 100$

(d) $N = 25$ (e) $N = 64$ (f) $N = 100$

Train on 25 node network with transmission range of 50m, generalize to 64 and 100 node networks with transmission ranges of 30 to 80m!

Summary and future work

Designed novel decentralized forwarding algorithm for mobile networks

Key ideas: Relational features, offline centralized training/online distributed testing, extended time action aka options

Future work:

- Mobile networks
- Flexibility in which packet in queue to send
- Super DeepRL strategy trained on multiple different scenarios
- Understanding extent of generalization ability

?? || /**/

vumanfredi@wesleyan.edu
https://victoriamanfredi.com

Backup